High Quality Math Editing & Display in Office 12

Murray Sargent III
Office Authoring Services

Overview

- 8 math infrastructures inside and outside of Microsoft enable better math display/editing
- New math edit/display environment
- Provide infrastructure for editing programs to support built-up formulas
- Provide ability to interchange formulae with popular mathematics programs such as Mathematica, MathCad and Matlab
- Incorporate into Word 12, OneNote 2.0, RichEdit, PowerPoint, IE, …

Overview

- 8 math infrastructures inside and outside of Microsoft enable better math display/editing
- New math edit/display environment
- Provide infrastructure for editing programs to support built-up formulas
- Provide ability to interchange formulae with popular mathematics programs such as Mathematica, MathCad and Matlab
- Incorporate into Word 12, OneNote 2.0, RichEdit, PowerPoint, IE, ...

Overview

- 8 math infrastructures inside and outside of Microsoft enable better math display/editing
- New math edit/display environment
- Provide infrastructure for editing programs to support built-up formulas
- Provide ability to interchange formulae with popular mathematics programs such as Mathematica, MathCad and Matlab
- Incorporate into Word 12, OneNote 2.0, RichEdit, PowerPoint, IE, ...

Collaborators

- Math layout handler: Eliyezer Kohen, Victor Kozyrev, Andrei Burago
- Word PM: Jennifer Michelstein
- New font: Mike Duggan, Geraldine Wade, Greg Hitchcock and Monotype
- OpenType: Sergei Malkin
- Devs/Test/PM of RichEdit, LineServices, Page/TableServices, Word, OneNote, Office Art
- Hand writing: MSR & MSRA
- Outside: Barbara Beeton, Asmus Freytag

Math Infrastructures

- [La]TeX: current tech-doc standards
- Unicode 4.0: includes ~2000 math symbols
- MathML 2.0: math K – 12 and beyond
- OpenType: special math tables
- New OpenType math font
- LineServices 4.0: math handler
- RichEdit 6.0: multilingual math edit engine
- MS Office environment, autocorrect

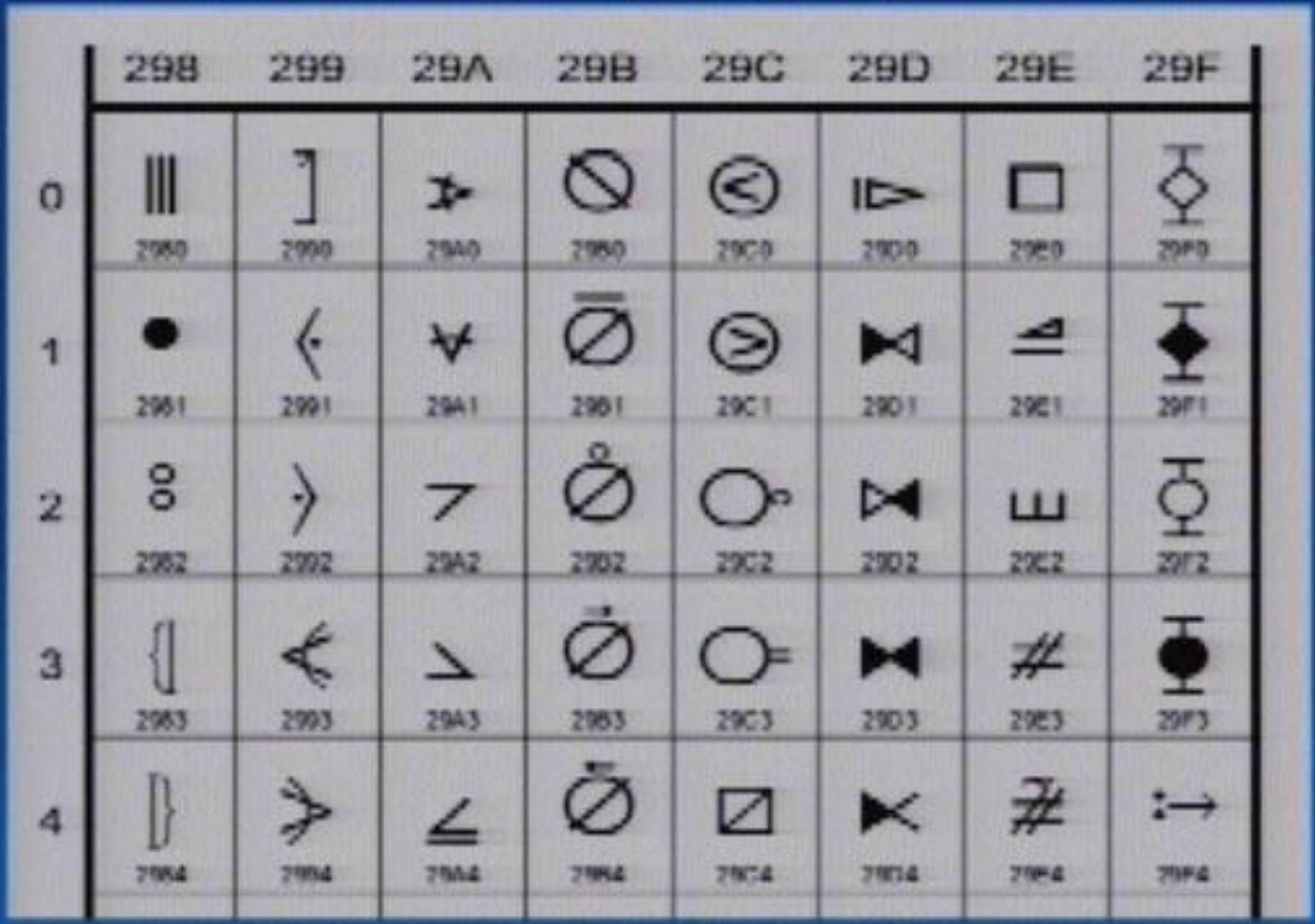
[La]TeX

- Widely used, high quality tech document preparation system
- Simple ASCII keyboard entry
- Usage and math typography are well documented
- Stable since 1986
- Complex scenarios are hard to manipulate
- Numerous dialects and macros complicate interchange

Unicode 4.0

- 340 math chars exist in ASCII, U+2200 block, arrows, combining marks
- 1016 math alphanumeric characters are in Unicode Plane 1 or Letterlike Symbols
- 591 new math symbols and operators are on BMP
- One math variant selector
- One new combining character (reverse solidus)
- New math characters were requested by STIX

Extensive Math Symbols



Basic Set of Alphanumeric Characters

- Latin digits (0 - 9)
- Upper- & lowercase Latin letters (a - z, A - Z)
- Uppercase Greek letters Α - Ω plus the nabla ∇ and the variant of theta ϴ given by U+03F4
- Lowercase Greek letters α - ω plus the partial differential sign ∂ and glyph variants of ϵ, ϑ, ϰ, φ, ϱ, and ϖ
- Only unaccented forms of letters are used

Math Alphanumeric Characters

- Math needs various Latin and Greek alphabets like normal, bold, italic, script, Fraktur, and open-face
- May appear to be font variations, but have distinct semantics
- Without these distinctions, you get gibberish, violating Unicode rule: *plain text must contain enough info to permit text to be rendered legibly, and nothing more*
- Plain-text searches should distinguish between alphabets, e.g., a search for script $\mathscr{H}$ shouldn't match H, etc.
- Reduces markup verbosity

Legibility Loss

Without math alphabets, the Hamiltonian formula

$$\mathcal{H} = \int d\tau \, [\varepsilon E^2 + \mu H^2]$$

becomes an integral equation

$$H = \int d\tau \, [\varepsilon E^2 + \mu H^2]$$

Unicode Plain Text

- Can do a lot with plain text, e.g., BiDi
- Grey zone: use of embedded codes
- Unicode ascribes semantics to characters, e.g., paragraph mark, right-to-left mark
- Lots of punctuation characters in range U+2000 to U+204F
- Extensive character semantics/properties tables, including mathematical, numerical
- Can use "plain text" for formula entry, searching

Unicode and Markup

- Unicode isn't intended to represent all aspects of text
- Language attribute: sort order, word breaks
- Rich (fancy) text formatting, character and paragraph format properties
- Content tags: headings, abstract, author, figure
- Glyph variants: Poetica font: 58 ampersands; Mantinia font: novel ligatures (TT, TE, etc.)
- MathML adds XML tags for math constructs
- TeX is also a markup language

MathML: For a better web

"MathML will make the Web even better for educational, scientific and technical materials. It also has the potential to make mathematics accessible to those with visual disabilities. It will allow mathematical content to be reused and exchanged with technical computing systems for further manipulation."

-- Tim Berners-Lee, W3C Director

MathML

- MathML 1.0 (April, 1998) was the first World Wide Web Consortium (W3C) endorsed XML vocabulary
- Low-level format for describing mathematics as a basis for machine to machine communication
- MathML facilitates the use and re-use of scientific content on the Web
- MathML 2.0 released in late 2003

MathML Presentation Markup

- Presentation markup directs how the math should be rendered.

```
<mrow>
    <mi>E</mi>
    <mo>=</mo>
        <mrow>
            <mi>m</mi>
              <mo>&InvisibleTimes;</mo>
              <msup>
                  <mi>c</mi>
                  <mn>2</mn>
              </msup>
        </mrow>
</mrow>
```

$$E = mc^2$$

MathML Content Markup

- Content markup describes the meaning of the expression, not the format.

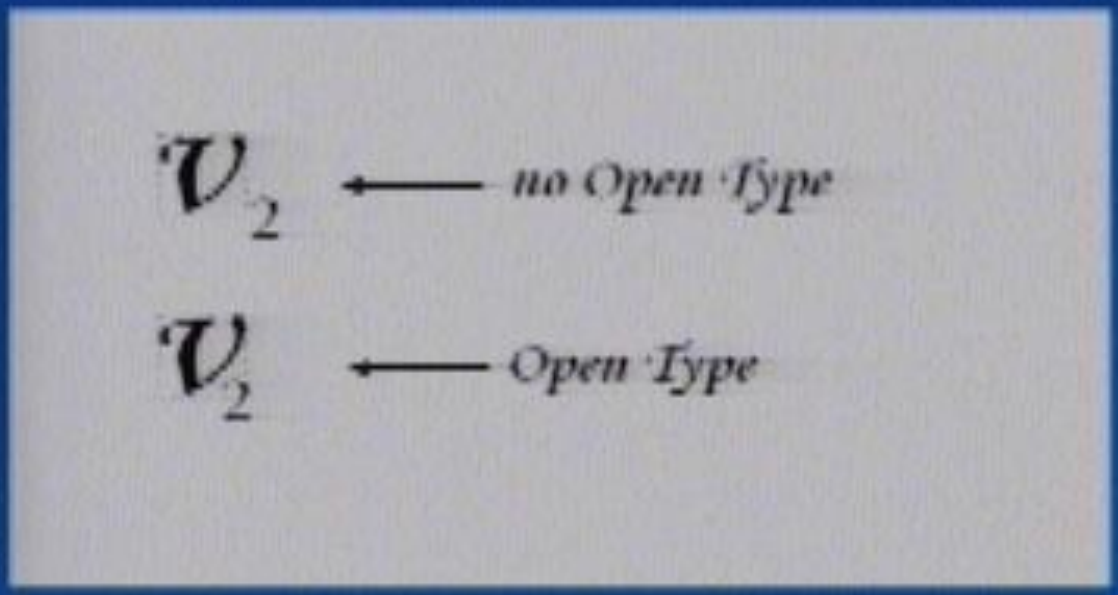
```
<mrow>
    <eq/>
    <ci>E</ci>
    <apply>
        <times/>
            <ci>m</ci>
            <apply>
                <power/>
                <ci>c</ci>
                <cn>2</cn>
            </apply>
        </times/>
    </apply>
</mrow>
```

$$E = mc^2$$

OpenType

- Specialized math tables can be created
- Position subscripts/superscripts horizontally better than TeX
- Reduce need for TeX "tweaking" with +tive and -tive thin spaces

Open Type Typography for WordEquations



New Math Font

- Designed by Jelle Bosma and produced by Agfa/Monotype
- Will contain math-specific OpenType tables and full Unicode math set
- Will be the default Word font, replacing Times New Roman
- Is designed with ClearType in mind
- STIX is also creating a new math font

LineServices 4.0

- Standard line layout component used by all major Microsoft text engines
- New math handler developed by Eliyezer Kohen, Victor Kozyrev, Andrei Burago
- In combo with Page/Table Services 3.0, enables optimal line breaking algorithm
- With OpenType and new font, can have better typography than TeX
- Automatic line breaking inside math zones

RichEdit 6.0

- RichEdit is set of plain/rich-text, single/multiline Unicode/ANSI edit controls and combo/listboxes in single world-wide binary
- Multilevel undo, message & com interfaces, considerable Word compatibility, rich formatting
- Outline view, zoom, font binding, latest in IME support, and rich complex script support (BiDi, Indic, and Thai)
- Rich ink support
- WYSIWYG editing of ruby, warichu, tatenakayoko, built-up math

Clients include

- Outlook mail note, post it, ...
- Most Office dialogs
- All OSes since Win98
  - Wordpad, Charmap
- Darwin installer
- WebCalc
- Project
- Visual Studio, DaVinci
- Publisher
- Front Page
- OneNote
- PowerPoint
- Handheld PC PocketWord
- eBooks
- OE (for mail header)
- Borland's Delphi
- SQL server dev tools, RAID
- MSN Companion chat
- Via Win2k Wrapper – cc:mail, WebEditPro, Eudora, Ericarla, Money(US), Sibelius, Borland TRichedit class, apps created with VB, MFC...
- ePeriodicals

Some Fancier Features

- Features added for ebooks: pagination, hyphenation, kerning, (subpixel) ClearType support, text wrap around embedded objects

Some Fancier Features

- Features added for ebooks: pagination, hyphenation, kerning, (subpixel) ClearType support, text wrap around embedded objects
- Multilevel tables

Some Fancier Features

- Features added for ebooks: pagination, hyphenation, kerning, (subpixel) ClearType support, text wrap around embedded objects
- Multilevel tables
- Autocorrect

Some Fancier Features

- Features added for ebooks: pagination, hyphenation, kerning, (subpixel) ClearType support, text wrap around embedded objects
- Multilevel tables
- Autocorrect
- AutoURL detection
- Features added for ePeriodicals: drop caps, multiple columns, OpenType support, optimal line breaking

Other Components used

- Uniscribe
- Line Services 4.0
- Page/Table Services 3.0
- Windows Text Services Framework
- Callbacks for access to word-break, auto correct, hyphenation, and ClearType libraries
- OLE
- mso

Components are accessed on demand

Other Components used

- Uniscribe
- Line Services 4.0
- Page/Table Services 3.0
- Windows Text Services Framework
- Callbacks for access to word-break, auto correct, hyphenation, and ClearType libraries
- OLE
- mso

Components are accessed on demand

Math Display/Editing Model

- Combination of 8 infrastructures
- Powerful input methods
- LineServices instead of OLE
- Equation numbering & multiline alignment
- Math user interface is an extension of app's UI
- Find/Replace is native in app
- Equation Editor & MathType use OLE…

Persistence

- MathML is preferred interchange format
- RTF's EQ field gives some interchange capability
- TeX is also desirable, but can be done via TeX ↔ MathML converters
- MathML allows easy interoperability with all major math programs, e.g., MathCad, Mathematica, Matlab, MathType

Input Methods

- TeX
- Autocorrect
- Streamlined linear format
- Autoformula build up
- Alt+x hexadecimal input
- Keyboards and menus
- Taskpane
- Handwritten formulas

TeX becomes cumbersome for longer equations such as

$$W^{3\beta}_{\delta_1\rho_1\sigma_2} - U^{3\beta}_{\delta_1\rho_1} + \frac{1}{8\pi^2}\int_{\alpha_1}^{\alpha_2} d\alpha_2' \left[\frac{U^{2\beta}_{\delta_1\rho_1} - \alpha_2' U^{1\beta}_{\rho_1\sigma_2}}{U^{0\beta}_{\rho_1\sigma_2}}\right].$$

The Unicode plain-text version of this reads as

$$W^{3\beta}_{\delta_1\rho_1\sigma_2} - U^{3\beta}_{\delta_1\rho_1} + 1/8\pi^2 \int_{\alpha_1}^{\alpha_2} d\alpha_2' \left[\frac{U^{2\beta}_{\delta_1\rho_1} - \alpha_2' U^{1\beta}_{\rho_1\sigma_2}}{U^{0\beta}_{\rho_1\sigma_2}}\right]$$

while the standard TeX version reads as

```
${W^{3\beta}_{\delta_1\rho_1\sigma_2}
    - U^{3\beta}_{\delta_1\rho_1} + {1 \over 8\pi^2}
    \int_{\alpha_1}^{\alpha_2} d\alpha_2' \left[
    {U^{2\beta}_{\delta_1\rho_1} - \alpha_2'
    U^{1\beta}_{\rho_1\sigma_2} \over
    U^{0\beta}_{\rho_1\sigma_2}} \right] }$
```

Unicode TEX Example

```
${W^{3\beta}_{\delta_1\rho_1\sigma_2}
= U^{3\beta}_{\delta_1\rho_1} + {1 \over 8\pi^2}
\int_{\alpha_1}^{\alpha_2} d\alpha_2\prime \left[
{U^{2\beta}_{\delta_1\rho_1} - \alpha_2\prime
U^{1\beta}_{\rho_1\sigma_2} \over
U^{0\beta}_{\rho_1\sigma_2}} \right] }$ .
```

In a "Unicoded" TEX, it could read as

```
${W^{3β}_{δ₁ρ₁σ₂} = U^{3β}_{δ₁ρ₁} + {1 / 8π²}
∫ {α₁}^{α₂} dα₂′ \left[ {U^{2β} {δ₁ρ₁} - α₂ U^{1β} {ρ₁σ₂}
/ U^{0β} {ρ₁σ₂}} \right] }$ .
```

Symbol Entry

- GUI PCs can display a myriad glyphs, mathematics symbols, and international characters
- Menu methods are slow, but intuitive. Hot keys are great but hard to learn. Menus can reveal hot keys
- Autocorrect greatly aids TeX-style input like \delta → δ
- With left/right Ctrl/Alt keys, PC keyboard gives direct access to 600 symbols. Maximum possible = $2^{100} = 10^{30}$
- Use on-screen, customizable, keyboards and symbol boxes
- Drag & drop any symbol into apps or onto keyboards

Hex to Unicode Input Method

- Type Unicode character hexadecimal code
- Make corrections as need be
- Type Alt+x to convert to character
- Type Alt+x to convert back to hex (useful especially for “missing glyph” character)
- Resolve ambiguities by selection
- Input higher-plane chars using 5 or 6-digit code
- MS Word and RichEdit standard

Formula AutoBuildUp

- Enter formulas in simple linear format
- When an expression is syntactically correct, it is automatically built up
- Edit expressions in built-up form or in linear form as desired
- For special operations like integral, type \int, (autocorrects to ∫) followed by subscript and superscript for limits, which auto build up

Linear format math

- Simple operand is a *span* of alphanumeric characters
- E.g., simple numerator or denominator is terminated by any nonalphanumeric character
- *abc/d* gives $\frac{abc}{d}$
- More complicated operands use parentheses ( ), brackets [ ], or { }
- Outermost parens aren't displayed in built-up form

Linear format math (cont)

- E.g., plain text (*a* + *c*)/*d* displays as $\frac{a+c}{d}$
- Easier to read than TEX's, e.g., {a + c \over d}
- MathML: <mfrac><mrow><mi>a</mi><mo>+</mo><mi>c</mi></mrow><mrow><mi>d</mi></mrow></mfrac>
- Neat feature: linear-format text looks like math

Subscripts and Superscripts

- Unicode has numeric subscripts and superscripts along with some operators (U+2070-U+208E)
- Others need some kind of markup like <msup> </msup>
- With special subscript and superscript operators (not yet in Unicode), these scripts can be encoded neatly
- Use TeX's _ and ^ subscript/superscript ops for now
- Use parentheses as for fractions to overrule built-in precedence order

Taskpane

- Categories of symbols for easy navigation
- Customizable categorization of symbols
- Recently Used Math Symbols (customizable)
- My Equations for equation storage and reuse (customizable)
- Single-click symbol insertion into document

Handwritten Equations

- Partner with MSR, MSRA for implementation
- Ink directly on Word document
- Select button to convert into text
- See next slide for example

I'm working on a problem set for my calculus class. Now I'm going to start writing equations by hand.

$$f(x) = \int_0^{\pi} \sin^2(x)\,dx$$

After selecting "convert," the handwriting is converted into text, as is illustrated below:

I'm working on a problem set for my calculus class. Now I'm going to start writing equations by hand.

$$f(x) = \int_0^{\pi} \sin^2(x)\,dx$$

Demo

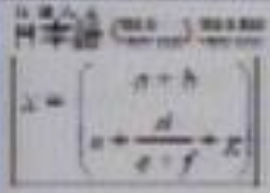
- RichEdit math displaying RTF file
- RichEdit autoformula buildup, autocorrect
- Paste into Word
- Save as XHTML with embedded MathML
- Display in IE with MathPlayer MathML behavior
- OneNote math

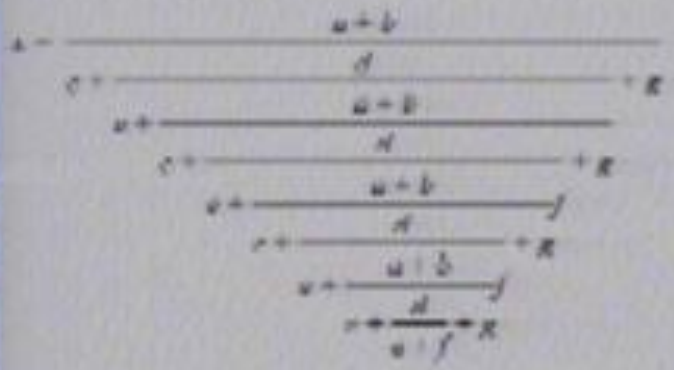
MS3 Built-up Math Programs

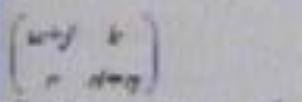
- *SCROLL - A pattern recording language*, AFIPS Conf. Proc. **35**:525-536 (1970)
- Snobol 4 version of SCROLL (1972)
- Daisy wheel controller (16KB) (1978)
- PS Technical Word Processor (1980-90)
- LineServices driver and RichEdit (2000 – present)

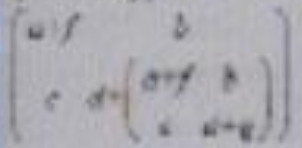
Where else?

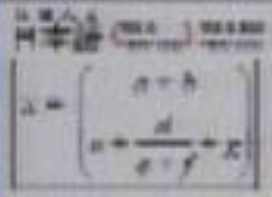
- Publisher could integrate math natively, since it uses LineServices, etc.
- Avalon text. But presumably not until next version
- Programming languages: wouldn't it be great if the formulas we program looked like their printed versions?

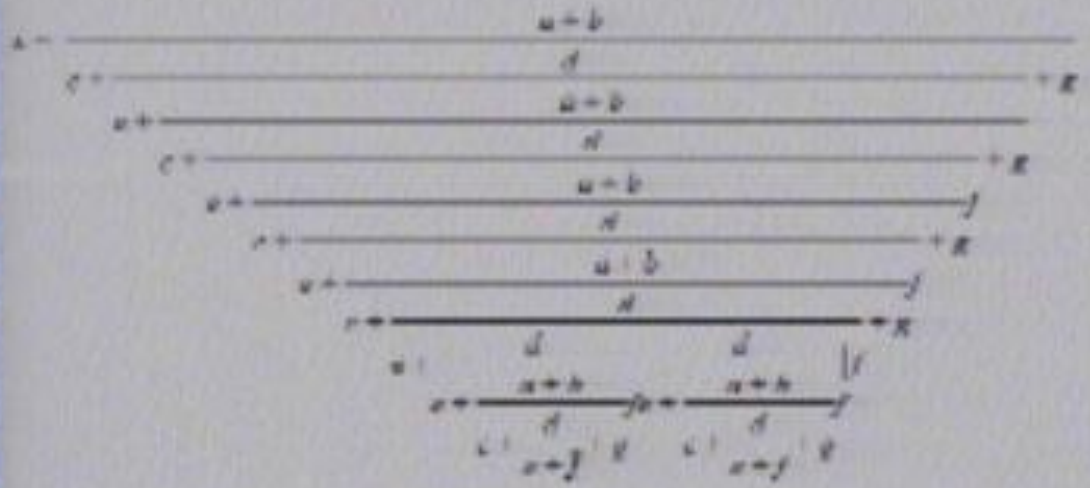


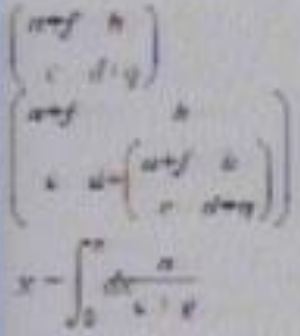


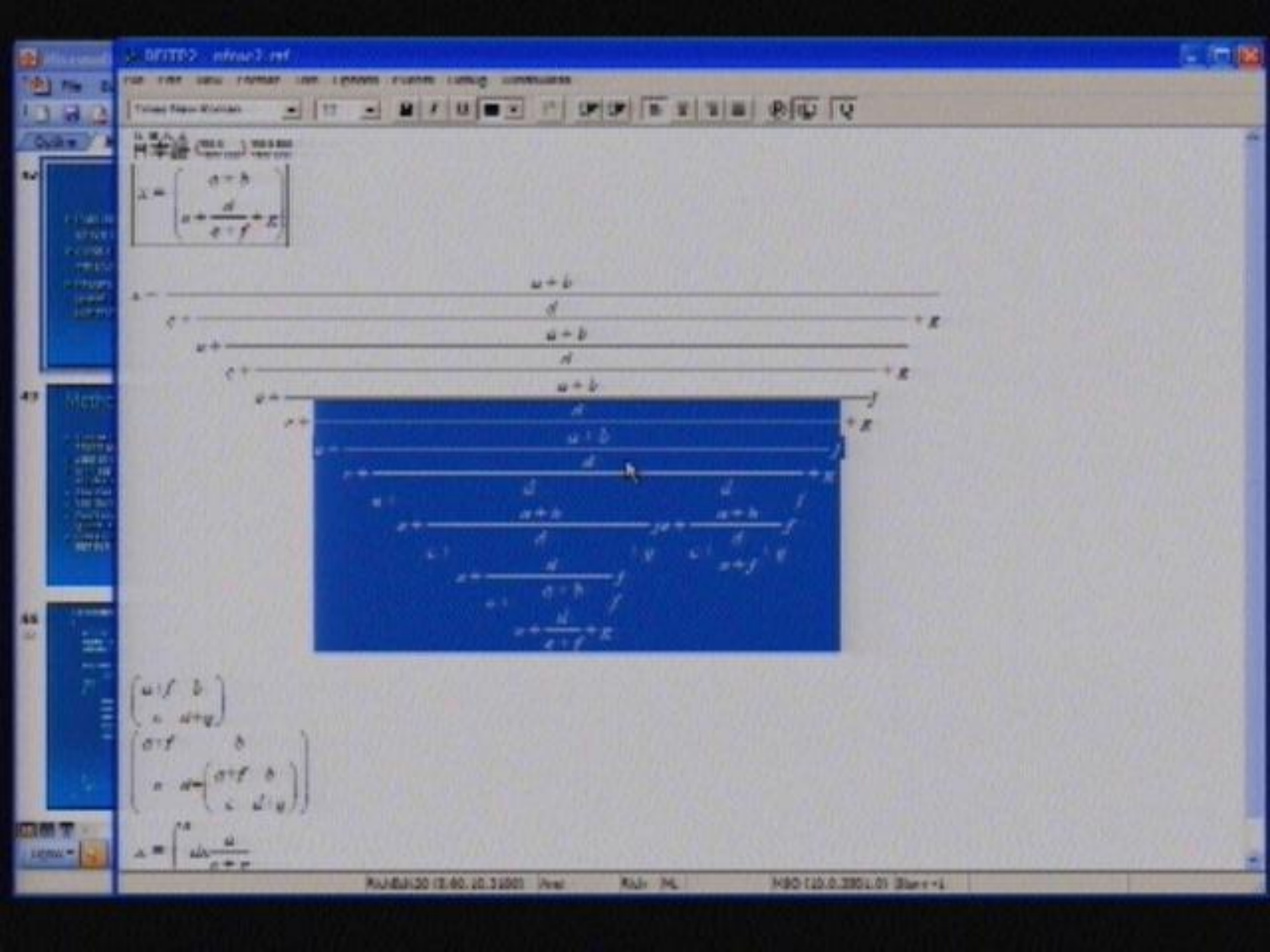


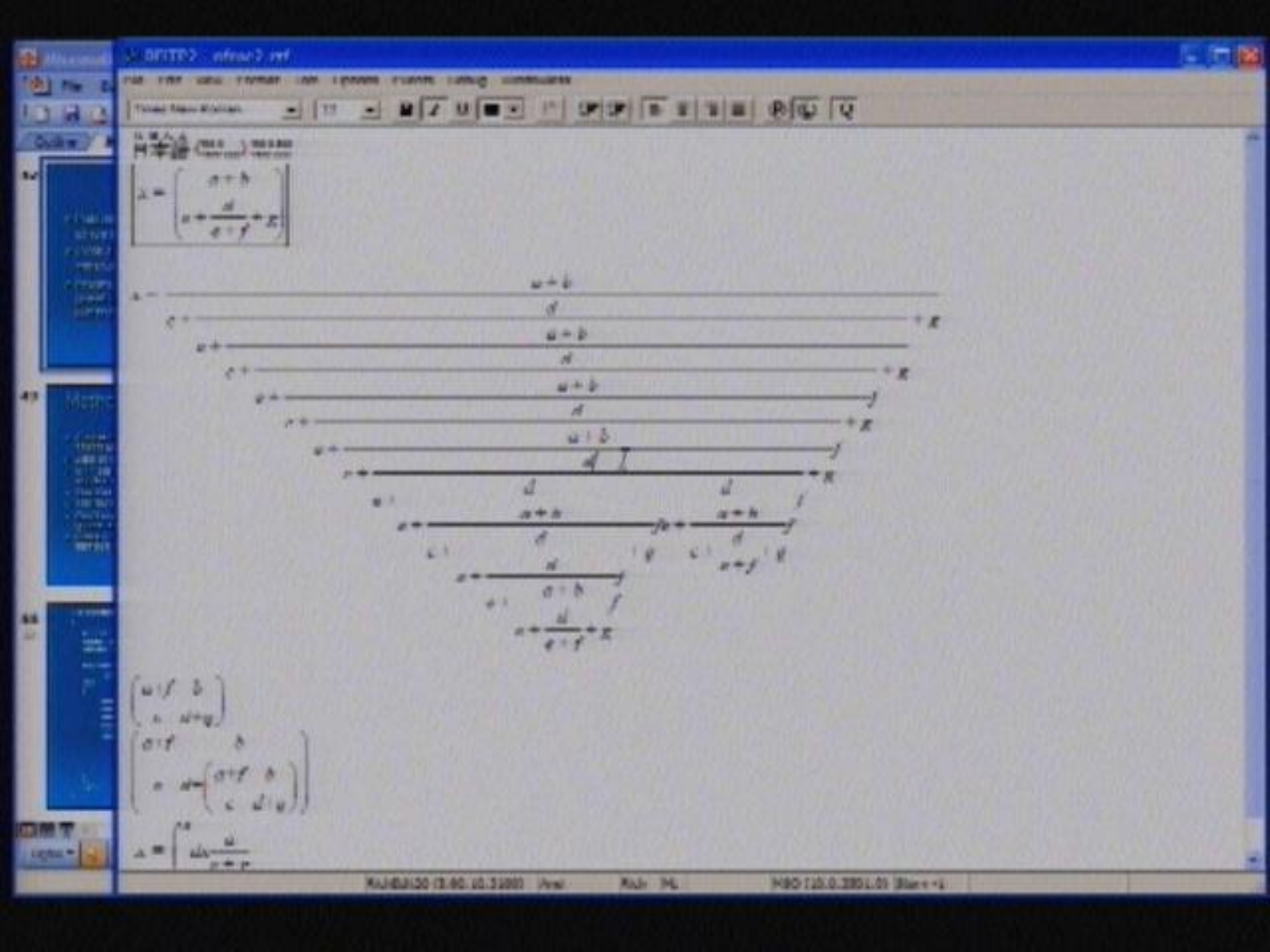


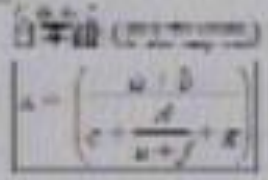




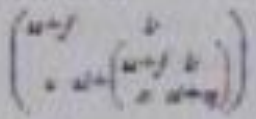


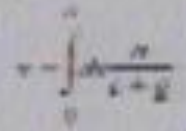


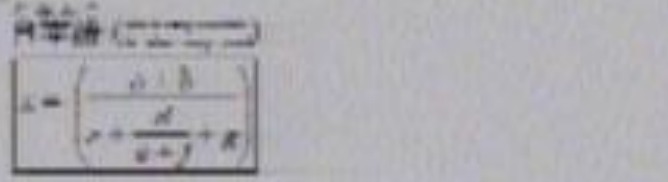


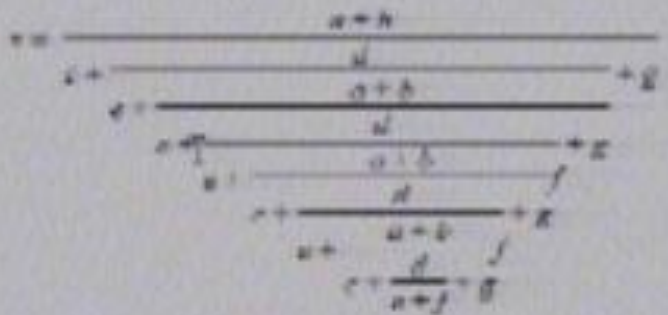


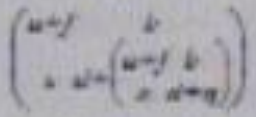
Kennel

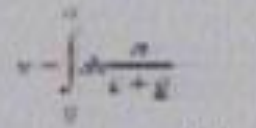


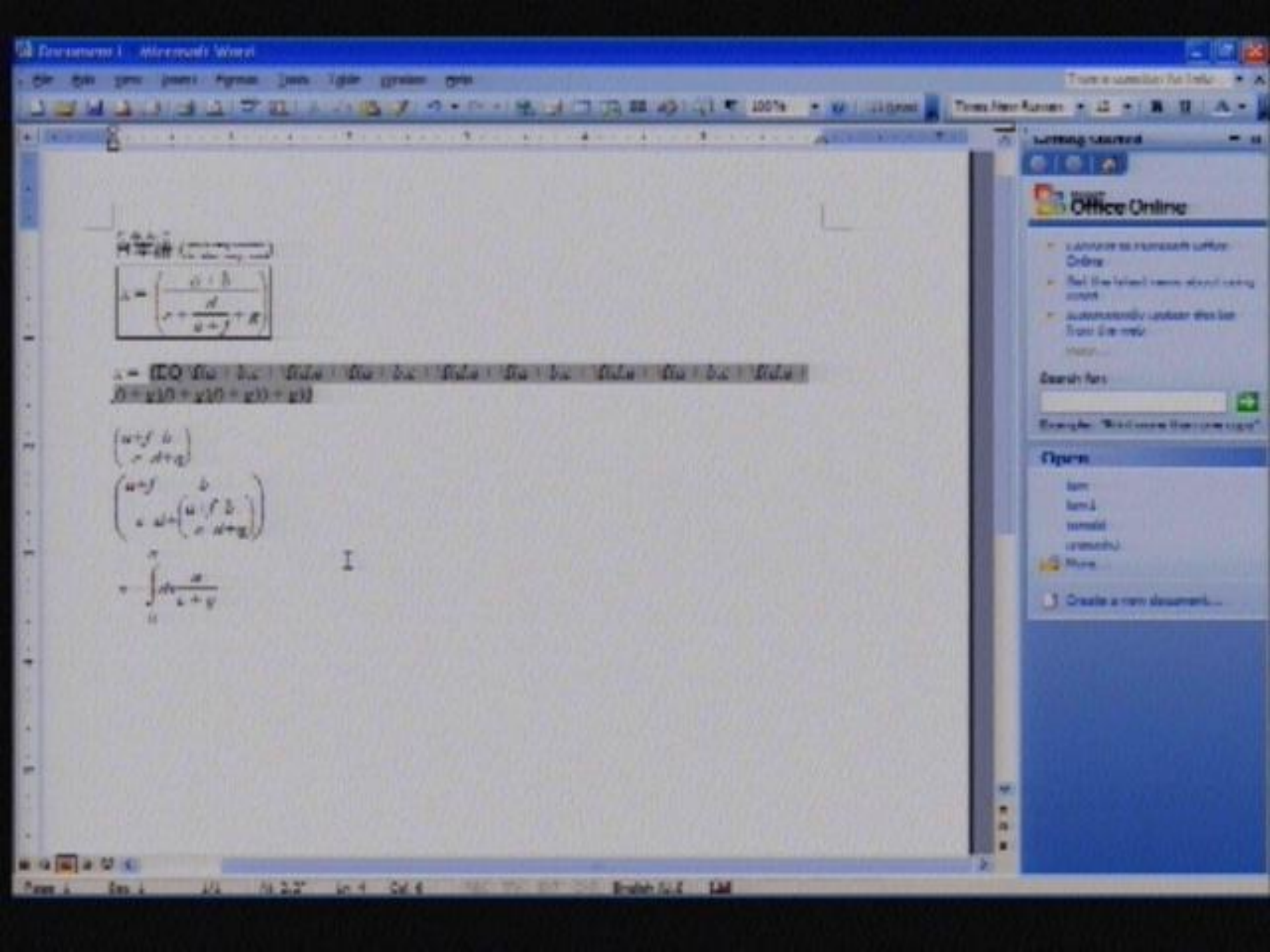


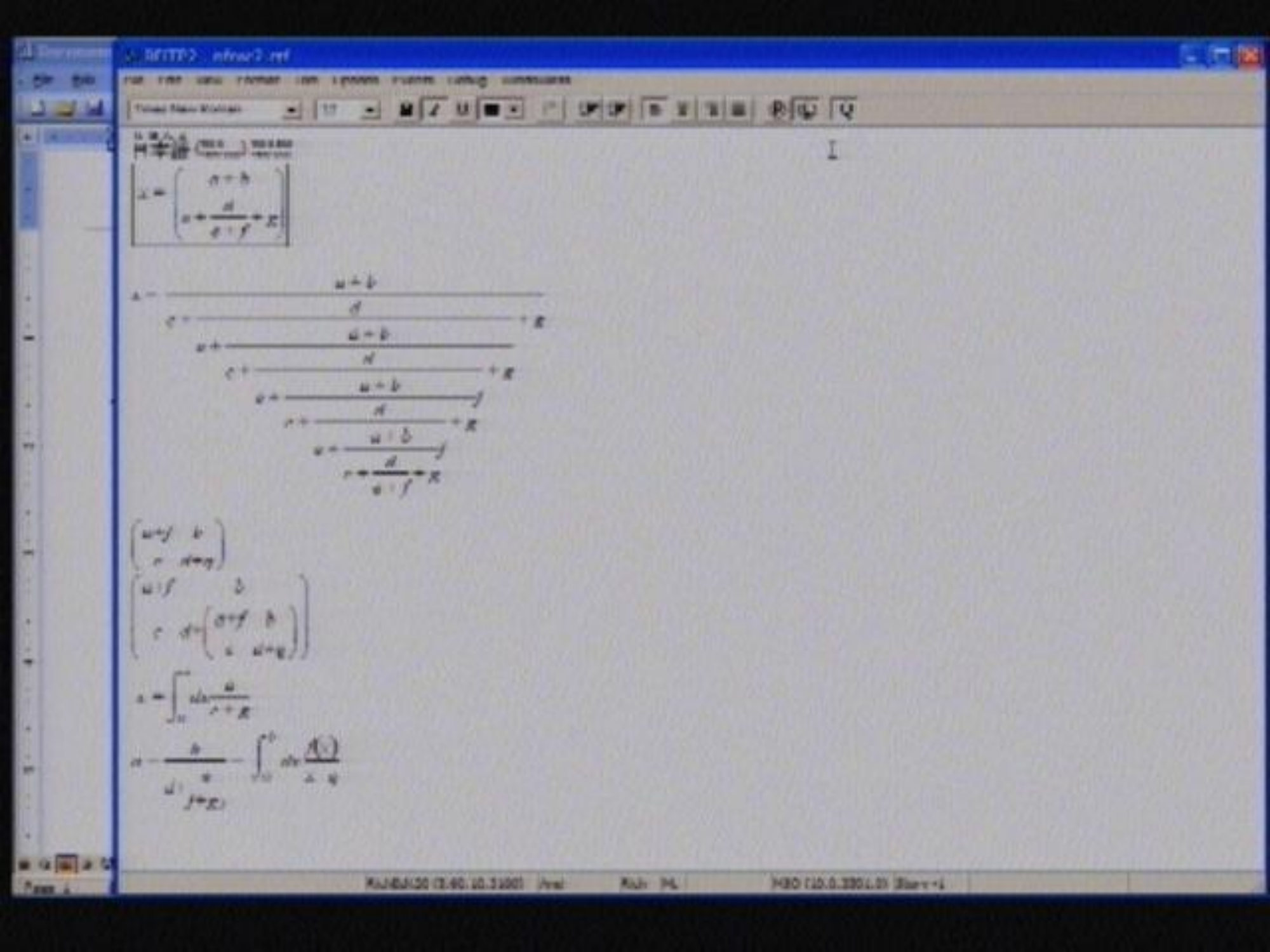


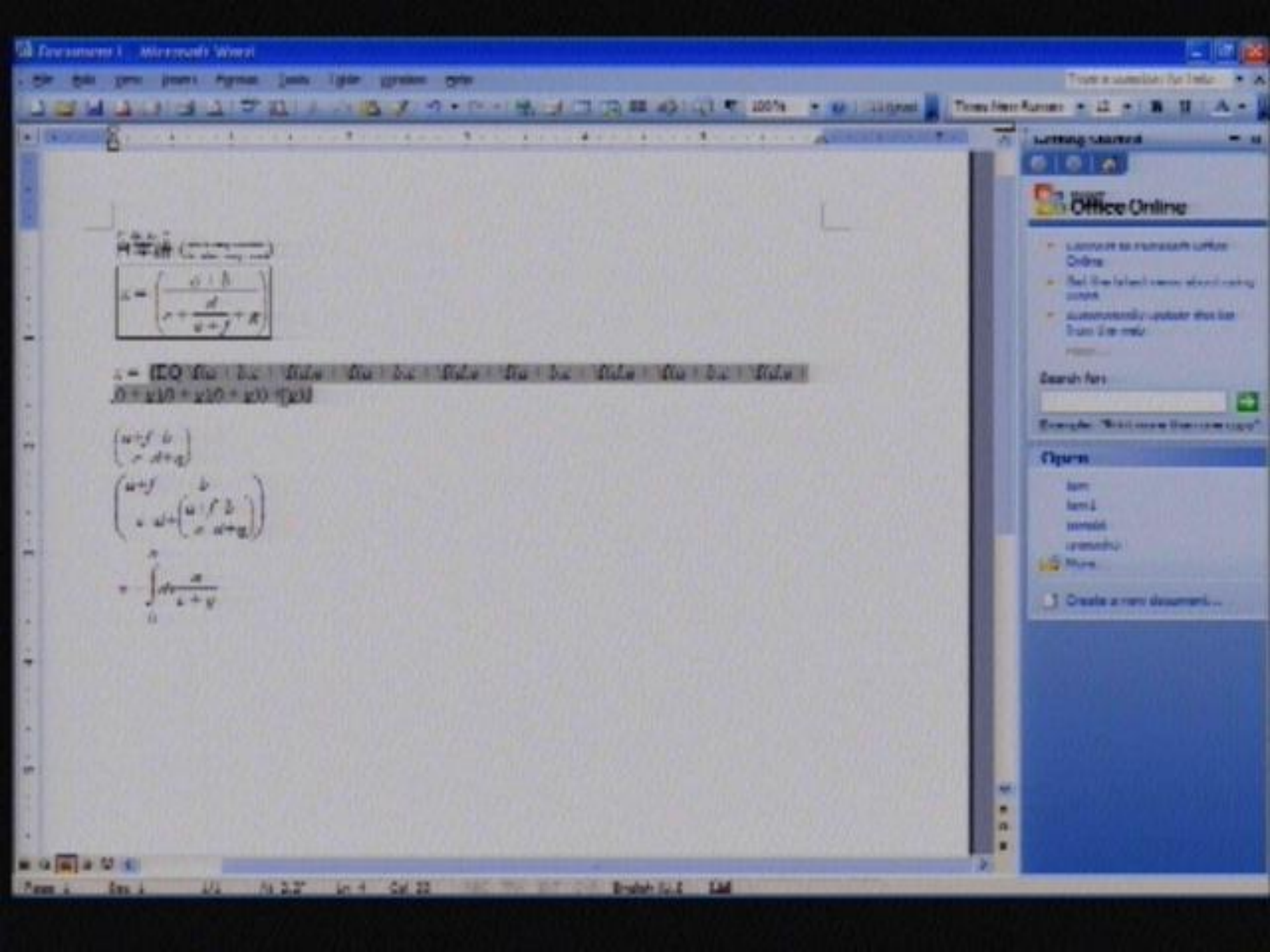


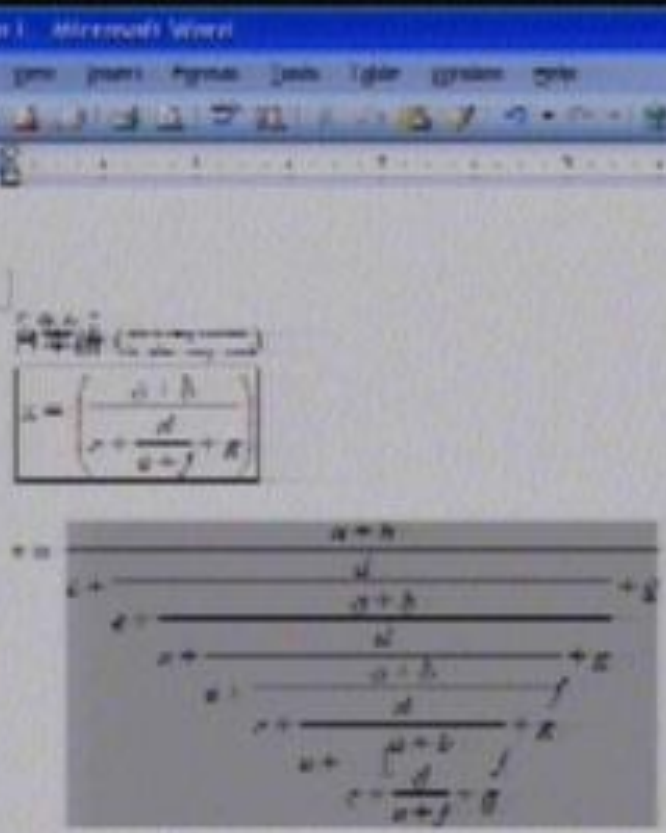


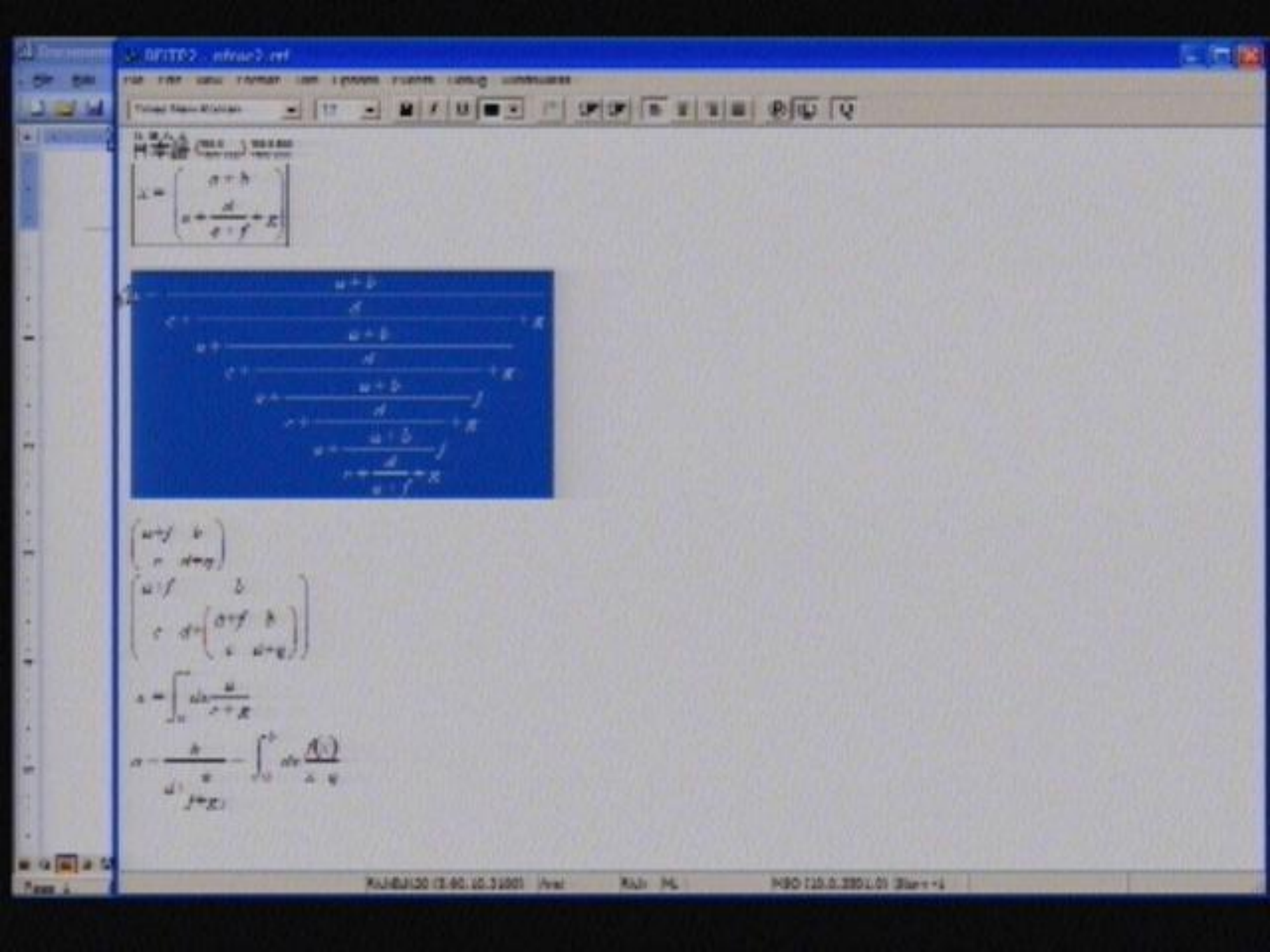


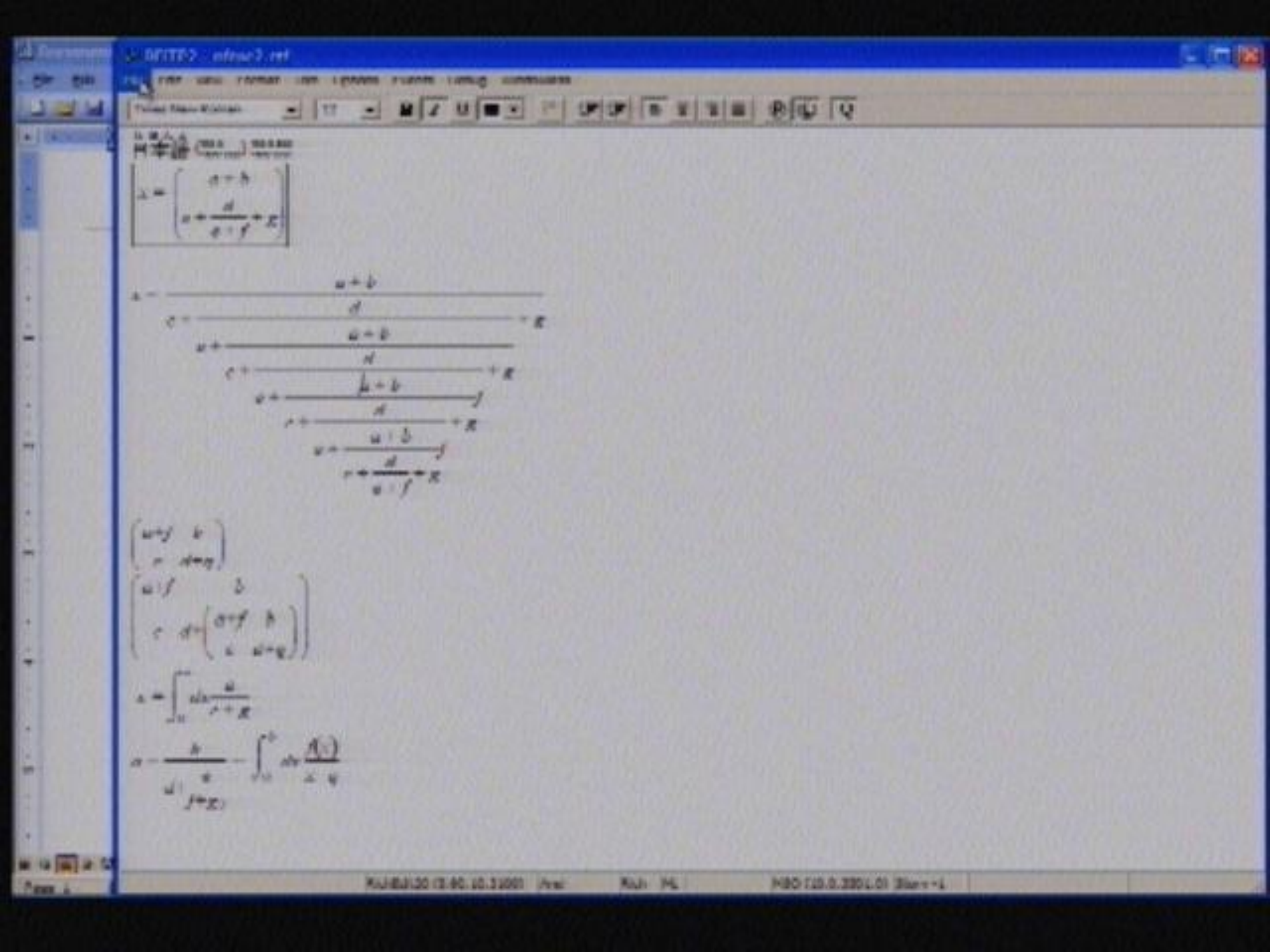


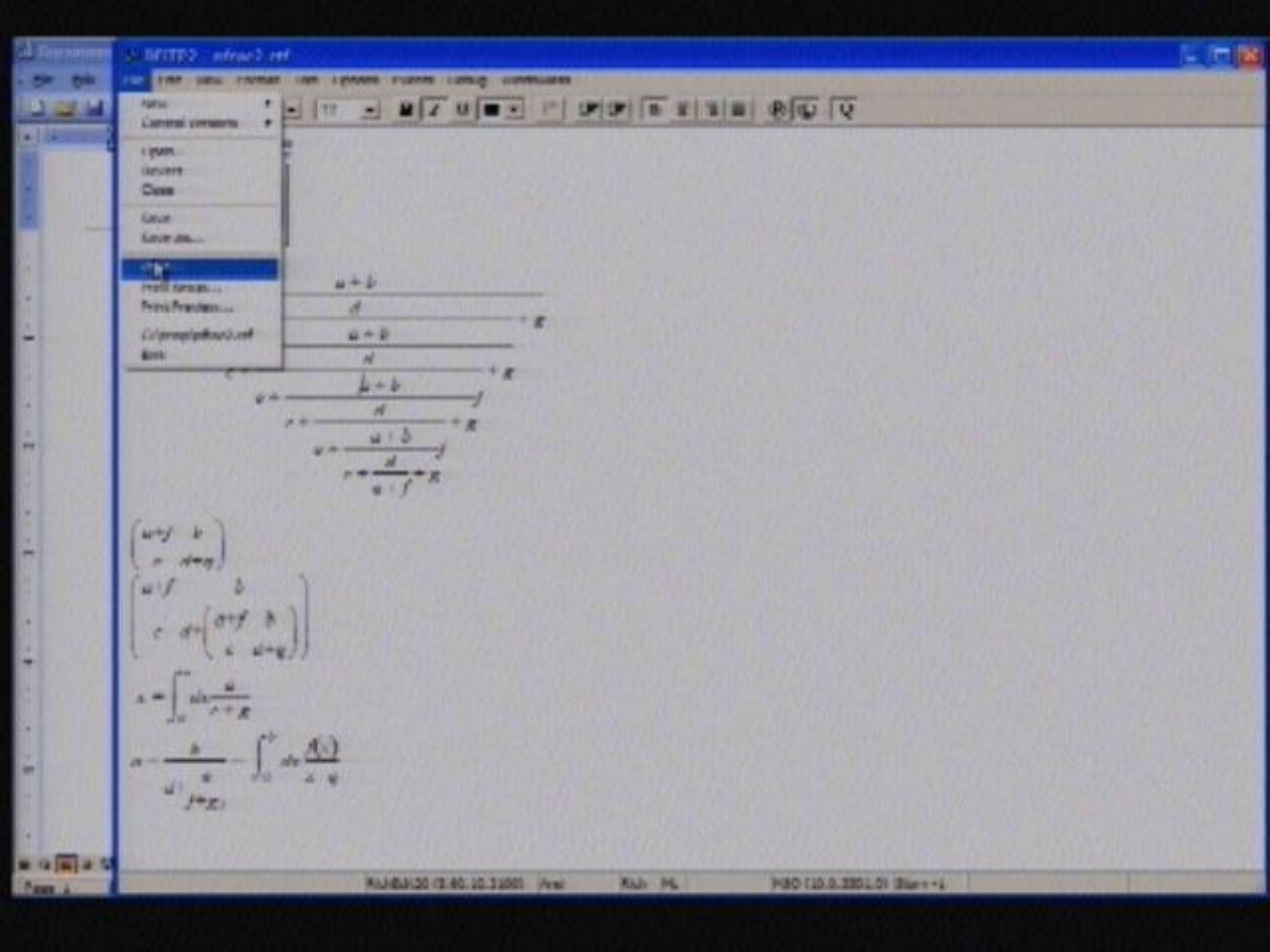


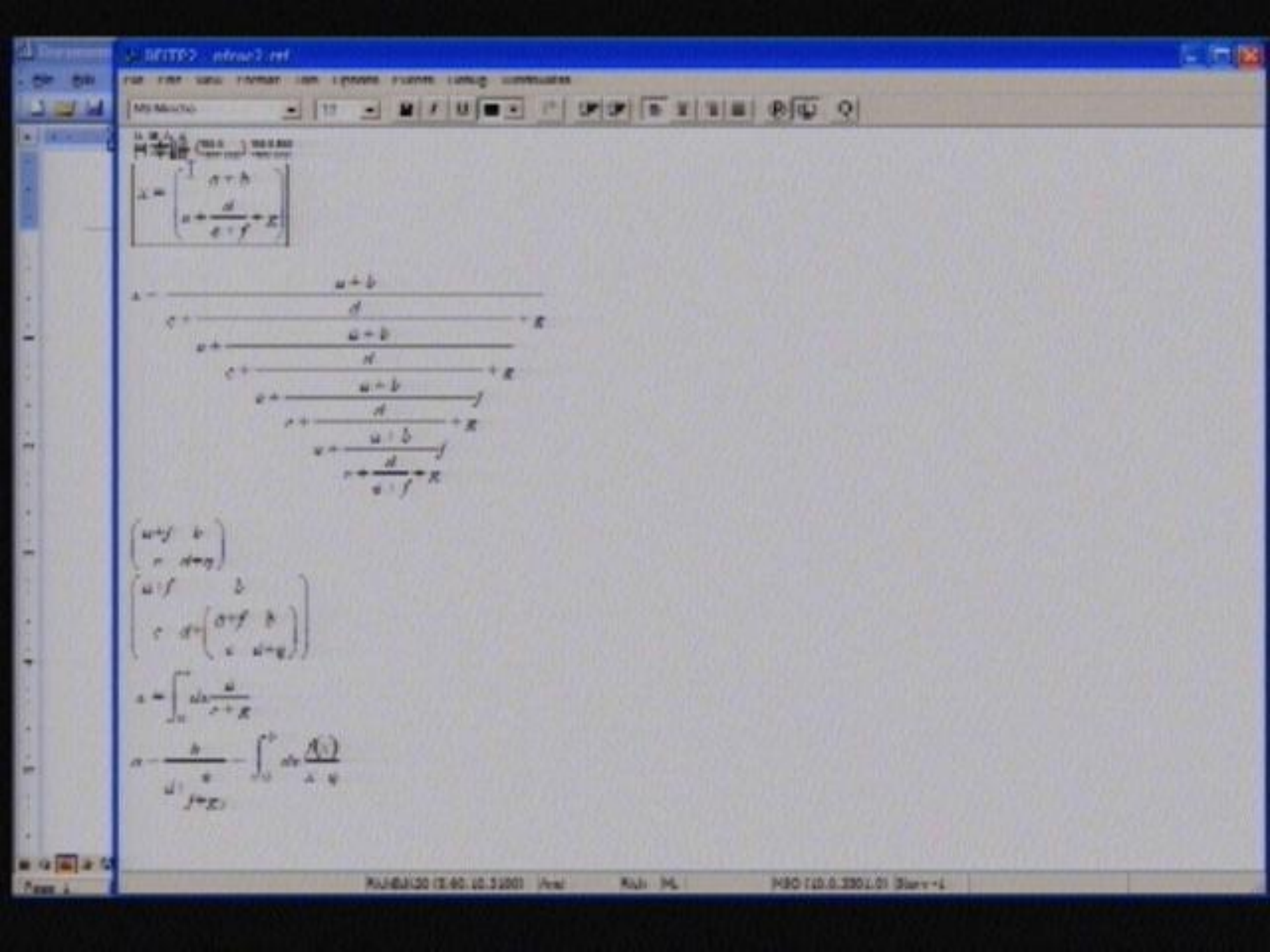












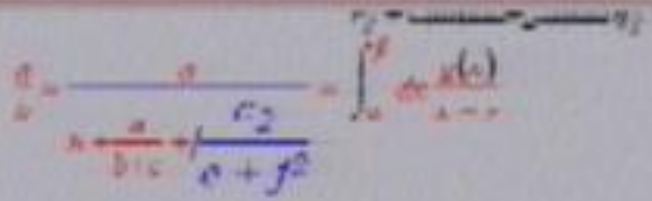
日本語 (this is very cool) this is also very cool

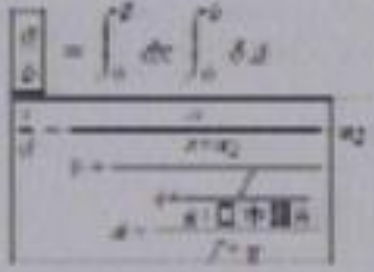
$$x = \left(\frac{a+b}{c + \frac{d}{e+f} + g} \right)$$

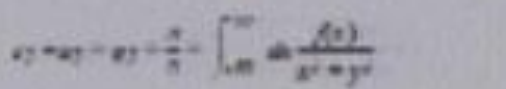
$$x = \cfrac{a+b}{c + \cfrac{d}{c + \cfrac{a+b}{e + \cfrac{d}{c + \cfrac{a+b}{e + \cfrac{d}{c + \cfrac{a+b}{e + \cfrac{d}{c + \cfrac{d}{e+f}} + g}} + g}} + f}} + g}$$

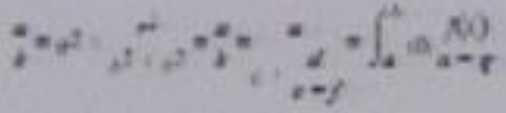
ab

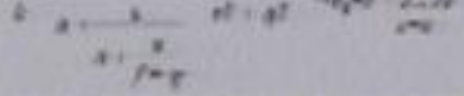
Friday, January 11, 2008

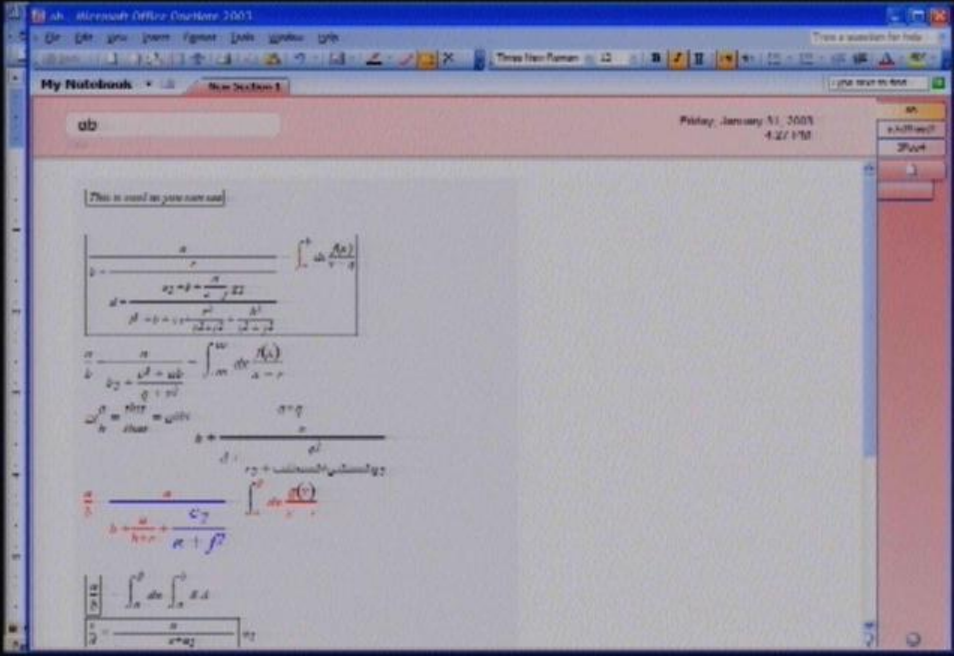












My Notebook
ab
Times New Roman

Where else?

- Publisher could integrate math natively, since it uses LineServices, etc.
- Avalon text. But presumably not until next version
- Programming languages: wouldn't it be great if the formulas we program looked like their printed versions?

Mathematics as a Programming Language

- Fortran made great steps in getting computers to understand mathematics
- Java and C# accept Unicode variable names
- C++ has preprocessor and operator overloading, but needs extensions to be really elegant for math
- Use Unicode characters including math alphanumerics
- Use linear encoding of mathematical expressions
- Can't use all mathematical expressions as code, but can go much further than current languages go
- When to multiply? In abstract, multiplication is infinitely fast and precise, but not on a computer

```
void IHBMWM(void)
{
    gammap = gamma*sqrt(1 + I2);
    upsilon = cmplx(gamma+gamma1, Delta);
    alphaeff = alpha0*(1-(gamma*gamma*I2/gammap)/(gammap + upsilon));

    if (!gamma1 && fabs(Delta*T1) < 0.01)
        alphacoh = -half*alpha0*I2*pow(gamma/gammap, 3);
    else
    {
        Gamma = 1/T1 + gamma1;
        I2sF = (I2/T1)/cmplx(Gamma, Delta);
        betap2 = upsilon*(upsilon + gamma*I2sF);
        beta = sqrt(betap2);
        alphacoh = 0.5*gamma*alpha0*(I2sF*(gamma + upsilon)
                   /(gammap*gammap - betap2))
                   *((1+gamma/beta)*(beta - upsilon)/(beta + upsilon)
                   - (1+gamma/gammap)*(gammap - upsilon)/
                   (gammap + upsilon));
    }
    alpha1 = alphaeff + alphacoh;
}
```

```
void IHBMWM(void)
{
        γ' = γ*√(1 + I2);
        υ = γ + γ1 + r*Δ;
        αinc = α0*(1 - (γ*γ*I2/γ')/(γ' + υ));
        if (!γ1 || fabs(Λ*T1) <= 0.01)
                αcoh = -.5*α0*I2*pow(γ/γ', 3);
        else
        {
                Γ = 1/T1 + γ1;
                I2F = (I2/T1)/(1 + r*Λ);
                β = √(β2 + υ*(υ + γ*I2F));
                αcoh = .5*γ*α0*(I2F*(γ + υ)/(γ'*γ' - β2))
                        *((1 + γ/β)*(β - υ)/(β + υ) - (1 + γ/γ')*(γ' - υ)/(γ' + υ));
        }
        α1 = αinc + αcoh;
}
```

```
void IHBMWM(void)
{
        γ' = γ*sqrt(1 + I2);
        υ = γ + γ1 + i*Λ;
        αinc = α0*(1 - (γ*γ*I2/γ'))/(γ' + υ);
        if (!γ1 || fabs(Λ*T1) > 0.01)
                αcoh = -.5*α0*I2*pow(γ/γ', 3);
        else
        {
                Γ = 1/T1 + γ1;
                I2F = (I2/T1)/(Γ + i*Δ);
                β = sqrt(β² - υ*(υ + γ*I2F));
                αcoh = .5*γ*α0*(I2F*(γ + υ))/(γ'*γ' - β²)*(((1 + χ/β)*(β - υ)/(β + υ)) - ((1 - χ/γ')*(γ' - υ)/(γ' + υ)));
        }
        α1 = αinc + αcoh;
}
```

References

- *The Unicode Standard, Version 4.0*, Addison-Wesley (2003)
- B. Beeton, A. Freytag, M. Sargent III, *Unicode support for mathematics*, http://www.unicode.org/reports/tr25/ (2003)
- D. E. Knuth, *The TeXbook*, Addison Wesley (1986)
- L. Lamport, *LaTeX*, Addison Wesley (1994)
- MathML 2.0 is documented at http://www.w3.org/Math/ (2003)

Conclusions

- 8 infrastructures allow us to do math display and editing better than ever
- High quality LineServices math handler enables better typography than TeX
- Streamlined input methods facilitate math entry
- Incorporate into Word 12, OneNote 2.0, RichEdit, PowerPoint, IE, … and maybe future compilers

; this is very cool ; this is also very cool

$$A = \left(\frac{a+b}{c + \frac{d}{e+f} + g} \right)$$

$$x = \cfrac{a+b}{d + \cfrac{e+f}{g + \cfrac{h+i}{j + \cfrac{k+l}{m + \cfrac{n+o}{p+q}}}}}$$

$$\begin{pmatrix} a+f & b \\ c & d+g \end{pmatrix}$$

$$\begin{pmatrix} a+f & b \\ c & d+\begin{pmatrix} a+f & b \\ c & d+g \end{pmatrix} \end{pmatrix}$$

$$\lambda = \int_0^\infty dx \frac{x}{1+x}$$